Comprehension Assistant for Languages of Baltic States

Inguna Skadiņa
Tilde
Vienības gatve 75a, Riga, Latvia
LV1004
inguna.skadina@tilde.lv

Andrejs Vasiļjevs
Tilde
Vienības gatve 75a, Riga, Latvia
LV1004
andrejs@tilde.lv

Daiga Deksne
Tilde
Vienības gatve 75a, Riga, Latvia
LV1004
daiga.deksne@tilde.lv

Raivis Skadiņš
Tilde
Vienības gatve 75a, Riga, Latvia
LV1004
raivis.skadins@tilde.lv

Linda Goldberga
Tilde
Vienības gatve 75a, Riga, Latvia
LV1004
linda.goldberga@tilde.lv

Abstract

This paper presents results of a pilot project for the development of a foreign text comprehension assistant. This tool provides word, phrase and simple sentence translation between the languages of the Baltic countries (Estonian, Latvian and Lithuanian) and widely used European languages (English, German, French and Russian). The paper presents the general architecture of the system, describes its main constituents and outlines difficulties in multilingual phrase translation. The system demonstrates original adaptation of rule based techniques and statistical methods to deal with language specificities, such as inflectional word forms, free word order, and the lack of sizeable, sufficiently representative parallel corpus.

1 Introduction

For relatively small languages such as languages of the Baltic countries, electronic dictionaries and comprehension assistance tools play an important role in communication. Until now, several commercial desktop electronic dictionaries have been developed. Most of them are bilingual (different bilingual dictionaries of Fotonia, Festart English-Latvian dictionary, English-Estonian dictionary by Filosoft, and others), some are multilingual (MOT GlobalDix by Kielikone, multilingual dictionaries of Tilde).

Although electronic dictionaries are useful for communication, they are insufficient to overcome language barriers. Even after finding a translation of each word in a sentence, the user is still left unaided to figure out which translations to choose and how to form a sentence from them. Translation of text units out of context is the main drawback of electronic dictionaries. The role of the word in a sentence or its part of speech are important in determining the right translation. Electronic dictionaries are also of little assistance in detecting idiomatic expressions. Even if an expression is provided in the dictionary, the user usually is not able to detect it in a source text and is mislead by a confusing word-by-word translation.

On the other hand, Machine Translation (MT) systems for larger languages are rapidly gaining global popularity. However, they are not able to

approach the quality of human translation. Therefore MT systems are appropriate for users with no or very limited language skills as a fast way of grasping the basic subject matter of the content.

An alternative solution is a comprehension assistant, which assists user in understanding of foreign language text (Feldweg and Breidt, 1996; Prószéky and Balázs, 2002; Deksne et al 2005). This approach addresses a usage scenario where the user has some knowledge of the target language but occasionally needs assistance in understanding unknown words or phrases. Users with intermediate language skills prefer to read the original text and use translation assistance only when it is necessary. The comprehension assistant provides possible translations of a phrase or a word in context, helps to understand the structure of the sentence or the phrase and find relations between words, detects and translates idiomatic expressions. Translation of phrases as well as possible translations of individual words are provided.

The translation is provided as a screen tip in the context of the source text. Users are not disturbed from the source text, they see the translation context, are involved in the translation process by translating incomprehensible phrases only and interpreting the text themselves.

We have generalized the above mentioned approach from a single language pair to multilingual approach, covering languages of the Baltic countries and the most popular European languages. The developed system architecture allows simple inclusion of new language pairs – since the major constituents are language independent, only the language dependent content needs to be filled for a new language pair.

2 System Architecture

The aim of the comprehension assistant is to identify individual phrases in the text and provide the user with full translation of the whole phrase, as well as separate translations of the words constituting the phrase.

The comprehension assistant is built from separate components, each of them having their own functionality. (See Figure 1).

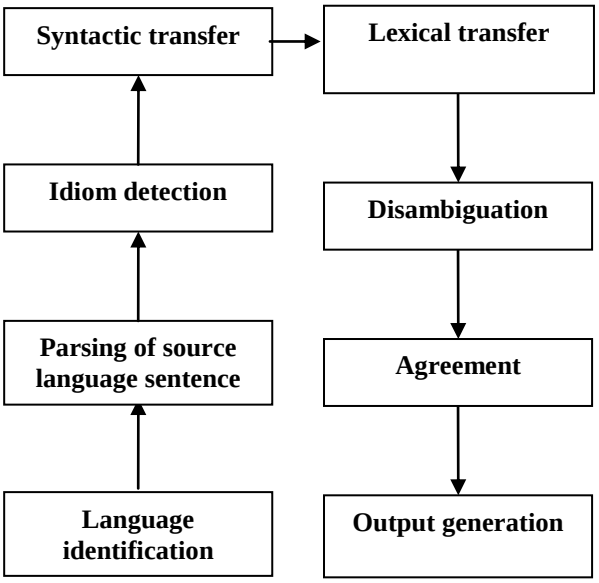


Figure 1. The chain of the comprehension assistant components.

During the translation process, components are executed successively. It means that the input data for each subsequent component are the internal structures created or processed by the previous component. At first, the system tracks the mouse pointer and retrieves text under it. Then it detects the language, analyzes the text, finds translation of the phrase containing the word under cursor and finds all translations for each word in the phrase. Finally, all results are presented to the user. Output contains both - the phrase translation and the translations of each word of the phrase. If the system cannot identify a phrase, translations of individual words are provided.

2.1 Language identification

Language identification module is developed to relieve the user from the need to select the translation source and target languages every time the language of the text changes. This module automatically identifies the language of the text and provides the appropriate source and target language information to the system. Currently the system identifies the following languages: English, Estonian, French, German, Latvian, Lithuanian and Russian.

For language identification, the character n-gram approach is used (Grefenstette, 1995; Bashir Ahmed et al, 2004). The *language reference model* is based on the most frequent character n-grams of sizes 1, 2, 3 and 4. For this pur-

pose the text corpus of every supported language is analyzed, most frequent sequences of one, two, three and four character long text strings are determined and probabilities of those n-grams are calculated.

During language identification of a particular text, we calculate frequency scores of character n-grams in this text to get the *text model*. The resulting text model is compared to the language reference models for all supported languages. The closest matching is based on 600 most characteristic n-grams of the language.

2.2 Parser

The aim of the parser component is to obtain a fully or partially parsed sentence. As the parsers differ from language to language, a wrapper component is developed, which transforms the output of different parsers to a unique format necessary for further processing. For widely spoken European languages, parsers are licensed from third party software vendors: Connexor[1], Dictum[2].

Parsers for Baltic languages have been developed within the project and have two constituents: the language independent parsing engine and the language dependent set of syntax rules.

The formal grammar we use for syntax rules is derived from unification grammar. Since Baltic languages are highly inflective languages, the syntax of the parsing rules needs to have attributes allowing inclusion of morphological information.

A parsing rule consists of two parts: description of the syntactic structure (a context free grammar rule) and usage conditions which describe constraints as well allow to assign or pass morphological and syntactic features between nodes.

In Figure 2, a simplified parser rule is shown. The rule describes the structure of a noun phrase (NP) consisting of an attributive adjective phrase (AP), the head noun (N) and an optional prepositional phrase (PP). The double equation mark '==' is used to describe conditions, i.e., the rule will be executed only if there will be agreement in case, gender and number between the adjective phrase (AP) and the noun (N). The single equation mark '=' is used to assign properties to the nodes. In the sample below, the noun phrase will inherit case, gender and number from the main noun.

```
NP -> attr:AP main:N (mod:PP)
      attr:AP.Case==main:N.Case
      attr:AP.Gender==main:N.Gender
      attr:AP.Number==main:N.Number
      NP.Case=main:N.Case
      NP.Gender=main:N.Gender
      NP.Number=main:N.Number
```

Figure 2. A simplified noun phrase parsing rule.

The parsing engine is based on CYK (Cocke-Younger-Kasami) algorithm (Cocke and Schwartz, 1970; Younger, 1967; Kasami, 1965). It uses bottom-up approach which allows partial parse of input sentence.

Original CYK algorithm supports context-free grammars written in Chomsky normal form (CNF). The developed rule formalism differs from CNF. Therefore parsing rules are transformed to CNF which is extended with attributes. The CYK parsing algorithm also was improved to handle attributes both for constraints and for assigning or passing attribute values between nodes.

Currently parsing rules are developed for Latvian and Lithuanian languages; for Estonian, small demo grammar is developed.

The output of the parser component is a syntax tree, or parts of the syntax tree of the sentence (see Figure 3) in case when full sentence parsing fails. Currently parsers for languages of the Baltic countries have no disambiguation constituent, therefore the first full parse tree, if it exists, is chosen for transfer. For the widely used European languages, parsers return a single parse tree.

[1] www.connexor.com
[2] http://www.dictum.ru/?main=products&sub=dictascope

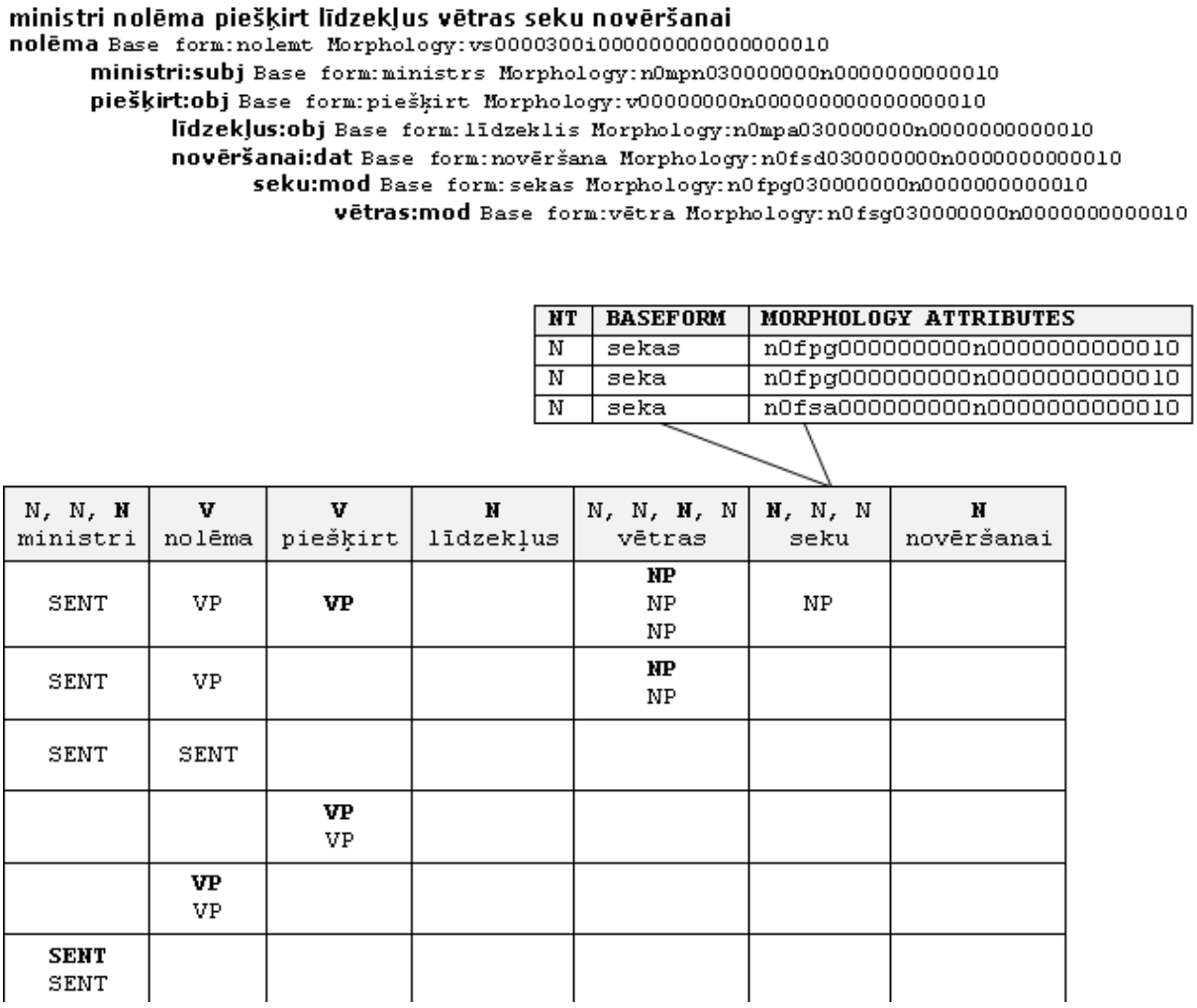


Figure 3. A parsed Latvian sentence in the form of the dependency tree (above) and as the matrix of the chunk parser (below).

2.3 Idiom processing

There are many cases in real texts where the meaning of a collocation of words is not based on the meaning of its parts. Baltic languages are not an exception and are rich in idiomatic expressions. For example, the literal translation of the Latvian expression *Gāž kā ar spaiņiem* (*It rains cats and dogs*) would be *Pouring like with buckets*.

Such idioms should be identified and treated as a whole in translation. In the comprehension assistant tool they are identified comparing adjacent words in the text to the stored list of idioms. If a matching idiomatic expression is found then the corresponding nodes in the parse tree are located and the translated idiom is attached to them. The information of the syntactic tree of the whole sentence is not used in idiom translation, however, the translated idiom is integrated into the tree to use it later in transfer, agreement and other processes.

Another specific case is translation of software interface elements. If the mouse pointer is located on menu items, the windows title bar, a dialog box message or other user interface elements, to increase quality of translation, specific dictionaries of pre-translated user interface strings and computer terminology are used.

The third case is English phrasal verbs which are language dependent (they are not typical for Latvian, Lithuanian and Russian) and are therefore handled in the syntactic transfer component.

2.4 Syntactic transfer

In the transfer phase, the syntactic tree in the source language is transformed into the corresponding syntactic tree for the target language.

Syntactical transformations are made to map one tree structure to another by applying transfer rules. The developed rule formalism allows to:

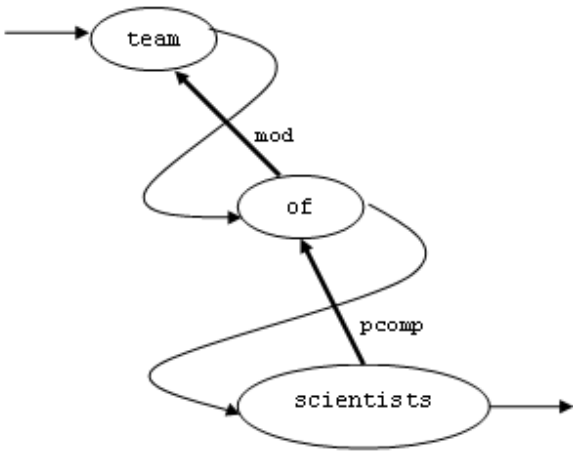
- change word order,
- delete or hide nodes,
- insert new nodes,
- transfer or assign syntactic, morphological or lexical properties,
- change type of syntactic relations between words.

Usually the transfer is applied to two or three syntactically related nodes, the order of which could be arbitrary in the text. Although transfer rules analyse syntactic relations between words, the word order could be changed during transfer. The following example shows a transfer rule for the transformation of a genitive phrase during translation from English into Latvian:

```
TransferRule(N<-mod-PREP<-pcomp-N)
{
        Child.SourceSpelling == ”of”;
        Grandchild.Case = genitive;
        MakeLink(Child – hidden -> Parent);
        Swap(GrandChild, Parent);
        MakeLink(GrandChild - mod -> Parent);
}
```

Figure 4. Transfer rule sample.

Applying this rule to the tree representing the English noun phrase ‘team of scientists’, the word ‘scientists’ will be moved to the position before the main word ‘team’ and the case of the word will be changed to the possessive case (genitive) and the preposition ‘of’ will be discarded.



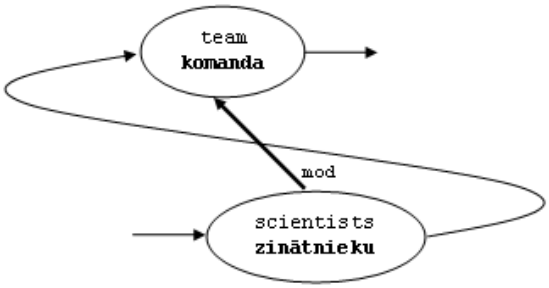


Figure 5. Sample syntactic tree before applying syntactic transfer rule (Figure 4) and after it. Light arrows show word sequence.

2.5 Lexical transfer

The lexical transfer component finds translations of the word in a bilingual dictionary based on the part of speech identified by the parser component. For example, for the English word *rest* in sentence *we need a rest*, noun translations (for Latvian: *atpūta, miers, pauze, pārtraukums*) will be selected and verb translations (for Latvian: *palikt, atpūsties, balstīties, gulties*) will be dismissed.

If there is no translation for the word in the required part of speech, the dictionary lookup is attempted for alternate classes. For instance, instead of a participle, the translation of adjective could be selected.

Usually, dictionaries include only translations of primary words without translations of derivations. For example, dictionaries usually have entries for words like ‘*assume*’, but less often they have entries for *‘assumption’, ‘assumed’* (adverb) or ‘*assuming*’ (noun), and they usually do not have entries for words like *‘assumer’* and *‘assumingly’*. For such cases, if the translation of a word is not in the dictionary, specific suffixes and prefixes are cut off at the end and the beginning of the word during dictionary lookup and added to the translated word of the target language. For example, a participle can be translated as the infinitive of the corresponding verb and then the required participle form is synthesized from the translation. Nouns can be cut off suffixes*: –tion, -er, -or*, then translated as verbs and the translations synthesized into the required nouns.

The obtained translations are arranged by their significance (score). Each translation has a label attached identifying whether it can be used in the translation of the phrase. Specific translations are not used in phrase translation, they appear only in the list for each word. In case when a single word is translated, the translations are taken from

a richer dictionary where translations are grouped by meanings, including comments on usage.

2.6 Disambiguation

The task of the disambiguation phase is to choose the most appropriate target language word from the several words selected in the lexical transfer phase. We use statistical methods for disambiguation. Traditionally bilingual corpus is used to get statistical data for disambiguation. For Baltic languages the available bilingual corpus is very limited, so we combined two approaches – using a monolingual corpus and multiword expressions with their translation equivalents extracted from the multilingual dictionary.

We applied different approaches for Latvian and Lithuanian. For Latvian disambiguation, we decided to take into account statistical data about the probability of syntactic pairs - two words being syntactically related in a phrase or sentence. This is a more advanced approach compared to bigram probability - probability of two words appearing next to each other in a sentence. We use several syntactic relations such as *subject(noun, verb)*, *object(verb, noun)*, *attribute(adjective, noun)* and *attribute(noun, noun)*.

We gathered a large corpus of Latvian texts from web content. We applied a shallow parser on this corpus to get pairs of syntactically related words. The frequency of each unique pair was calculated. Frequency data were normalized to get probability of syntactic pairs. We call the resulting data the *syntactic language model* (SLM) and use it for disambiguation.

In the syntactic tree of the target language we have one or more Latvian language words mapped to every node (source language word). For every connected Latvian word pair in the tree we find probability from the Latvian SLM. Now we can disambiguate the syntactic tree by selecting those translations that give the highest probability for the whole tree representing the phrase or the sentence.

This SLM based disambiguation improves the quality of the translation compared to the most primitive method of using just the first translation from the dictionary. But the drawback of this method is usage of target language data only and ignoring the source language text in disambiguation.

For Lithuanian disambiguation, we tried a more advanced approach. We used an English-Lithuanian dictionary with a large number of phrase translations. We applied shallow parsing to it and aligned Lithuanian syntactic bigrams with the corresponding English syntactic bigrams. Again the frequency and probability of such bilingual pairs were calculated. We call the resulting data the *syntactic translation model* (STM).

For English-Lithuanian translation, we find probability in the Lithuanian syntactic tree for every combination of English source and Lithuanian target words at one node connected with the same combination at other node. Probability for this bilingual pair (EN/LT –EN/LT) is found in the English-Lithuanian STM.

Usage of the STM model should potentially provide improved disambiguation quality than the SLM model. But we realized that for quality improvements we need much larger bilingual corpus of phrase translations than we have from the English-Lithuanian dictionary we used. Currently, the SLM model demonstrates better results but another comparison should be performed after creating a larger bilingual corpus and rebuilding STM.

As seen in Figure 6, different translations of the verb "pick" are chosen when it is used with nouns ’berries’, ’gift’ and ‘nose’.

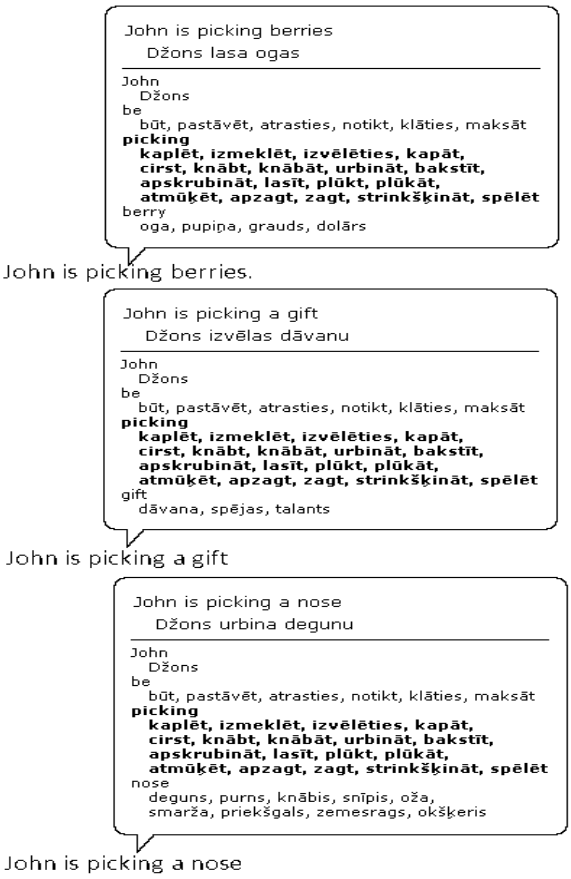


Figure 6. Disambiguation of meanings of the word ‘pick’ in English-Latvian translation.

2.7 Agreement

At the end of the disambiguation process, the target language syntactic tree contains only one target language word at each tree node. Tree nodes have some morphological properties (e.g., tense for verbs, case and number for nouns) set during parsing and transfer phases. But there are just target language dependent properties which must be set depending on the properties of other words and syntactic relations of words in the target language. For example, in the Baltic languages, the noun and the adjective must agree in case, number and gender. This agreement is established by agreement rules.

```
Rule(N<-attr-A)
        {
        Child.Number = Parent.Number;
        Child.Case = Parent.Case;
        Child.Gender = Parent.Gender;
        }
```

Figure 7. Agreement rule which assigns adjective (A) child node properties of parent noun node (N): gender, case and number.

Through agreement rules, the agreement module passes properties from one word to other and sets the missing morphological properties so that all morphological properties are set and all words in the phrase are in agreement.

Finaly, word form generation is applied according to the morphological properties of the word.

2.8 Output generation

The last phase is formatting of the resulting phrase or sentence.

The module returns translation results to the user according to the current position of the mouse pointer on the source text. The largest translated phrase related to the selected source word is returned together with translations of separate words of the phrase.

Figure 8. Translation from Lithuanian into English.

3 Achieved results and future work

Currently the comprehension assistant is at the stage of a pilot project – all system components are implemented and dictionaries for all language pairs are included. However, the level of phrase/sentence translation differs for different language pairs – currently it is better developed for Baltic languages (Latvian, Lithuanian) and less developed for Estonian. For Estonian, currently only small grammar is developed, and a rich set of Estonian syntax rules for this system is being currently implemented. Also English and Russian translation directions are more developed while for German and French only the basic syntactic constructions are currently implemented.

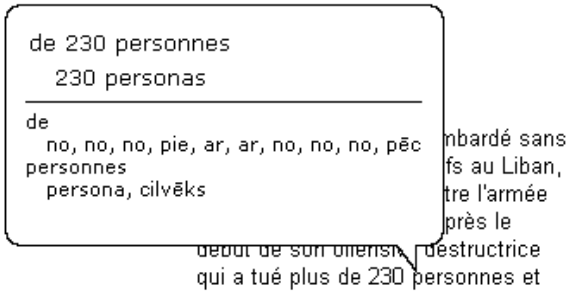


Figure 8. French-Latvian phrase translation.

Quality of translation of phrases varies depending on the complexity of the text. The system can handle relatively simple phrases, but fails dealing with texts from specific domains or dealing with texts with complex grammar and idiomatic meaning, like news headlines.

For test purposes, the gold standard for each language pair is developed. It contains main syntactic constructions for each language pair, as well as some typical cases of word sense disambiguation are included. Tests of the system have shown several weaknesses of the system. This is the basis for future work on improvement of the system.

One of the problems is proper nouns which are not distinguished, therefore, they sometimes are translated with a standard dictionary and the obtained translation does not match the context. In future, we should improve the functionality of proper noun recognition and they should be identified and translated using special dictionaries.

There is still a lot of work to be done to improve the quality of the dictionaries. To improve

translation quality, a revised dictionary is necessary which would meet usage-specific criteria.

Quality of dictionaries is important but dictionaries alone can not solve ambiguity issues. The disambiguation algorithm should be improved and statistic data (syntactic translation model) for disambiguator should be gathered from the large scale parallel corpus.

During development, system tests on the gold standard are performed; in future, evaluation of the whole system is planned.

References

Bashir Ahmed, Sung-Hyuk Cha, and Charles Tappert. 2004. Language Identification from Text Using N-gram Based Cumulative Frequency Addition, *Proceedings of Student/Faculty Research Day, CSIS, Pace University*.

Cocke John, Schwartz Jacob T. 1970. Programming languages and their compilers: Preliminary notes. *Technical report, Courant Institute of Mathematical Sciences*, New York University.

Grefenstette Gregory. 1995. Comparing two Language Identification Schemes, *JADT 1995, 3rd International conference on Statistical Analysis of Textual Data, Rome.*

Deksne Daiga, Skadiņa Inguna, Skadiņš Raivis, Vasiļjevs Andrejs. 2005. Foreign language reading tool – first step towards English-Latvian commercial machine translation system. *Proceeding of the Second Baltic Conference on Human Language Technologies*, Tallinn, 113-118.

Feldweg Helmut, Breidt Elisabeth. 1996. COMPASS - An Intelligent Dictionary System for Reading Text in a Foreign Language. *Papers in Computational Lexicography (COMPLEX 96)*, Linguistics Institute, HAS, Budapest, 53-62.

Kasami T. 1965. An efficient recognition and syntax-analysis algorithm for context-free languages. *Scientific report AFCRL-65-758*, Air Force Cambridge Research Lab, Bedford, MA.

Prószéky, Gábor. 2002. Comprehension Assistance Meets Machine Translation. Tomaš Erjavec; Jerneja Gros (eds) *Language Technologies*, 1–5. Institut Jošef Stefan,Ljubljana, Slovenia.

Prószéky Gábor, Balázs Kis. 2002. Development of a Context-Sensitive Dictionary. *Proceedings of the 10th International Congress of the European Association for Lexicography (EURALEX)*, Vol. I, 281–290. Copenhagen, Denmark.

Younger Daniel H. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control 10(2)*, 189–208.

Vasiļjevs Andrejs, Ķikāne Jana, Skadiņš Raivis. 2004. Development of HLT for Baltic languages in widely used applications. *Proceedings of First Baltic Conference „Human Language Technologies – the Baltic Perspective"*, Riga, 198-201.